

# EXERCÍCIOS

1. Crie uma classe chamada "Círculo" que possua um atributo para armazenar o raio e métodos para calcular a área e o perímetro do círculo.
2. Implemente uma classe chamada "ContaBancária" que possua atributos para armazenar o número da conta, nome do titular e saldo. Adicione métodos para realizar depósitos e saques.
3. Crie uma classe chamada "Retângulo" que possua atributos para armazenar a largura e a altura. Implemente métodos para calcular a área e o perímetro do retângulo.
4. Implemente uma classe chamada "Aluno" que possua atributos para armazenar o nome, a matrícula e as notas de um aluno. Adicione métodos para calcular a média das notas e verificar a situação do aluno (aprovado ou reprovado).
5. Crie uma classe chamada "Funcionário" com atributos para armazenar o nome, o salário e o cargo do funcionário. Implemente métodos para calcular o salário líquido, considerando descontos de impostos e benefícios.
6. Implemente uma classe chamada "Produto" que possua atributos para armazenar o nome, o preço e a quantidade em estoque. Adicione métodos para calcular o valor total em estoque e verificar se o produto está disponível.
7. Crie uma classe chamada "Triângulo" com atributos para armazenar os três lados do triângulo. Implemente métodos para verificar se é um triângulo válido e calcular sua área.
8. Implemente uma classe chamada "Carro" com atributos para armazenar a marca, o modelo e a velocidade atual do carro. Adicione métodos para acelerar, frear e exibir a velocidade atual.
9. Crie uma classe chamada "Paciente" que possua atributos para armazenar o nome, a idade e o histórico de consultas de um paciente. Implemente métodos para adicionar uma nova consulta ao histórico e exibir as consultas realizadas.
10. Implemente uma classe chamada "Livro" com atributos para armazenar o título, o autor e o número de páginas do livro. Adicione métodos para emprestar o livro, devolvê-lo e verificar se está disponível.
11. Implemente uma classe chamada "Banco" que represente uma instituição financeira. Essa classe deve conter métodos para cadastrar clientes, abrir contas bancárias e realizar operações como saques, depósitos e transferências.
12. Crie uma classe chamada "LojaVirtual" que represente uma plataforma de vendas online. Essa classe deve ter funcionalidades para cadastrar produtos, gerar carrinho de compras, aplicar descontos e calcular o valor total da compra.
13. Implemente uma classe chamada "Agenda" que represente uma agenda telefônica. Essa classe deve permitir adicionar, editar e remover contatos, além de buscar por contatos a partir de um nome ou número de telefone.
14. Crie uma classe chamada "MáquinaDeVendas" que simule uma máquina de venda de produtos. Essa classe deve permitir cadastrar produtos, selecionar um produto para compra, inserir dinheiro, retornar o troco e exibir o estoque disponível.
15. Implemente uma classe chamada "JogoCartas" que represente um jogo de cartas simples, como o Uno. Essa classe deve ter métodos para embaralhar as cartas, distribuir as cartas aos jogadores e permitir jogadas.

**SENAI "Roberto Mange"**

16. Crie uma classe chamada "RedeSocial" que represente uma rede social online. Essa classe deve ter funcionalidades para adicionar amigos, publicar mensagens, comentar em posts e buscar por usuários.
17. Implemente uma classe chamada "Biblioteca" que represente uma biblioteca virtual. Essa classe deve permitir cadastrar livros, fazer empréstimos, devolver livros e verificar a disponibilidade de um livro.
18. Crie uma classe chamada "Calendario" que represente um calendário anual. Essa classe deve ter métodos para exibir o calendário de um determinado mês, verificar se uma data é feriado e calcular a diferença de dias entre duas datas.
19. Implemente uma classe chamada "JogoAdivinhacao" que represente um jogo de adivinhação. Essa classe deve gerar um número aleatório, permitir que o jogador faça palpites e informar se o palpite está correto, informando se é maior ou menor que o número gerado.
20. Implemente um jogo de tabuleiro em que cada peça (peões, torres, etc.) seja representada por uma classe, com regras específicas de movimento e captura. No caso do xadrez, adicione a regra de roque, en passant, promoção de peões e xeque-mate.
21. Construa uma aplicação de e-commerce onde produtos, clientes e pedidos são objetos. Adicione funcionalidades como carrinho de compras, desconto, cálculo de frete, histórico de compras e recomendação de produtos com base nas compras anteriores.
22. Implemente um sistema para gerenciar um estoque de produtos com classes como Produto, Categoria, Fornecedor, Compra, Venda. Inclua funcionalidades de controle de quantidade, preço de compra, preço de venda e relatórios de inventário.
23. Crie um sistema de gerenciamento de tarefas onde você tem tarefas com prioridades, datas de vencimento, status (pendente, em andamento, concluída) e categorias. Implemente funcionalidades de criar, editar, listar e excluir tarefas, além de filtros por status e prioridade.
24. Desenvolva um sistema de gestão de um zoológico, com classes para Animal, Habitat, Alimentação, Veterinário, e Funcionário. Cada tipo de animal terá características específicas, como dieta e necessidades ambientais, e o sistema deve permitir o controle de alimentação, saúde e movimentação dos animais.
25. Crie um jogo de combate onde personagens são objetos com atributos como saúde, força, defesa, e habilidades especiais. Implemente combate entre personagens, uso de itens (como poções de cura), e eventos de vitória ou derrota. O sistema pode incluir também níveis de experiência e evolução dos personagens.

# DESAFIO

## 1. Sistema de Gerenciamento de Eventos

**Descrição:** Implemente um sistema de gerenciamento de eventos, como um software para planejar e organizar eventos corporativos, casamentos, festas, etc. O sistema deve gerenciar o agendamento de eventos, as tarefas associadas a cada evento, os fornecedores e a distribuição de responsabilidades.

**Classes principais:** Evento, Tarefa, Fornecedor, Responsável, Convidado, Pagamento..

**Requisitos:**

- Cada **evento** tem uma data, hora, local e descrição. Ele pode ter várias **tarefas** associadas (ex: aluguel de espaço, contratação de comida, etc.).
- As **tarefas** têm um responsável, prazo e um status (pendente, concluída, atrasada).
- O sistema deve ter um controle de **fornecedores** (como serviços de catering, decoração, segurança), permitindo associar fornecedores aos eventos.
- **Pagamentos:** O sistema deve permitir o controle de pagamentos relacionados ao evento (parcela, valor total, data de vencimento).

Desafios adicionais:

- **Polimorfismo** para diferentes tipos de eventos, como casamentos, eventos corporativos ou festas, com regras e exigências diferentes para cada tipo.
- **Consultas e relatórios:** Implemente funções para filtrar eventos por data, status de tarefa ou fornecedor, e gerar relatórios de status de pagamentos e tarefas.